

# Performance Guaranteed Approximation Algorithm for Minimum $k$ -Connected $m$ -Fold Dominating Set

Zhao Zhang<sup>1</sup> Jiao Zhou<sup>1</sup> Xiaohui Huang<sup>1</sup> Ding-Zhu Du<sup>2</sup>

<sup>1</sup> College of Mathematics Physics and Information Engineering, Zhejiang Normal University  
Jinhua, Zhejiang, 321004, China.

<sup>2</sup> Department of Computer Science, University of Texas at Dallas  
Richardson, Texas, 75080, USA

## Abstract

To achieve an efficient routing in a wireless sensor network, connected dominating set (CDS) is used as virtual backbone. A fault-tolerant virtual backbone can be modeled as a  $(k, m)$ -CDS. For a connected graph  $G = (V, E)$  and two fixed integers  $k$  and  $m$ , a node set  $C \subseteq V$  is a  $(k, m)$ -CDS of  $G$  if every node in  $V \setminus C$  has at least  $m$  neighbors in  $C$ , and the subgraph of  $G$  induced by  $C$  is  $k$ -connected. Previous to this work, approximation algorithms with guaranteed performance ratio in a general graph were known only for  $k \leq 3$ . This paper makes a significant progress by presenting a  $(2k - 1)\alpha_0$  approximation algorithm for general  $k$  and  $m$  with  $m \geq k$ , where  $\alpha_0$  is the performance ratio for the minimum CDS problem. Using currently best known ratio for  $\alpha_0$ , our algorithm has performance ratio  $O(\ln \Delta)$ , where  $\Delta$  is the maximum degree of the graph.

**Keyword:** wireless sensor network; virtual backbone; connected dominating set; fault-tolerance; approximation algorithm.

## I. INTRODUCTION

A wireless sensor network (WSN) consists of large quantities of sensors, which collaborate to accomplish a task. In implementing a WSN, saving energy is an important issue. If all sensors participate in the transmission frequently, it is clearly a waste of energy. Furthermore, such an active participation in transmission will incur a more serious problem known as *broadcast storm*, which is the result of intense interferences. Motivated by these considerations, people proposed using *virtual backbone* to play the main role of transmission [6], [8], which can be modeled as a connected dominating set in a graph.

Given a connected graph  $G = (V, E)$ , a node set  $C \subseteq V$  is a *dominating set* (DS) of  $G$  if every node in  $V \setminus C$  has at least one neighbor in  $C$ . It is a *connected dominating set* (CDS) of  $G$  if  $G[C]$  (the subgraph of  $G$  induced by  $C$ ) is connected. Using CDS as virtual backbone, information can be shared in the whole network. On the other hand, since the number of nodes participating transmission is reduced, the energy is saved and the interference is lessened.

In many applications, fault-tolerance is desirable. The concept of  $(k, m)$ -CDS is used to describe a fault-tolerant virtual backbone [5]. A node set  $C$  is a  *$k$ -connected  $m$ -fold dominating set* ( $(k, m)$ -CDS for abbreviation) if every node in  $V \setminus C$  has at least  $m$ -neighbors in  $C$  and  $G[C]$  is  $k$ -connected. Using a  $(k, m)$ -CDS as the virtual backbone, in the presence of at most  $\min\{m-1, k-1\}$  faults, information can still be shared in the remaining network.

#### A. Related Works

Connected dominating set was proposed in [6], [8] to serve as a virtual backbone of a wireless sensor network for the purpose of saving energy and alleviating interference. Guha and Khuller [10] proved that a minimum CDS cannot be approximated within a factor of  $(1 - \varepsilon) \ln n$  for any  $0 < \varepsilon < 1$  unless  $NP \subseteq DTIME(n^{O(\log \log n)})$ . In the same paper, Guha and Khuller gave two approximation algorithms achieving performance ratios of  $2(H(\Delta) + 1)$  and  $H(\Delta) + 2$ , respectively, where  $\Delta$  is the maximum degree of the graph and  $H(\cdot)$  is the Harmonic number. Using a new method dealing with non-submodular potential functions, Ruan *et al.* [17] improved the ratio to  $\ln \Delta + 2$ .

A homogeneous wireless sensor network can be modeled as a *unit disk graph*, in which every node of the graph corresponds to a sensor on the plane, and two nodes are adjacent in the graph if and only if the Euclidean distance between their corresponding sensors is at most one unit. For the minimum CDS problem in a unit disk graph, Cheng *et al.* [4] proposed a PTAS, which was generalized by Zhang *et al.* [29] to unit ball graph in higher dimensional space. These are centralized algorithms. Distributed algorithms can be found in [11], [12], [22], [23], [27]. Comprehensive studies on the algorithmic aspect of CDS is collected in the monograph [7].

Dai and Wu [5] were the first to propose the problem of constructing fault-tolerant virtual backbone. They presented three heuristics for  $(k, k)$ -CDS, without giving theoretical analysis on the performance ratio. After that, a lot of works appear in this field. The state of art studies are summarized in Table I. Since we are now talking about fault-tolerant virtual backbone,

at least one of  $k$  and  $m$  is assumed to be at least 2. Except for [24], all other results in this table assumes  $m \geq k$ . It can be seen from Table I that when  $k \leq 3$ ,  $(\ln \Delta + o(\ln \Delta))$ -approximation exists for  $(k, m)$ -CDS in a general graph, which is asymptotically best possible because of the inapproximability of this problem [10]. As to  $(k, m)$ -CDS on unit disk graph, constant approximation exists when  $k \leq 3$ . Recently, we [20] and Fukunage [9] independently achieved constant ratio for the minimum *weight*  $(k, m)$ -CDS problem on a unit disk graph when  $m \geq k$ . As far as we know, there is no performance guaranteed approximation algorithm for the minimum  $(k, m)$ -CDS problem on a general graph.

### B. Our Contribution

In this paper, we present an approximation algorithm for the minimum  $(k, m)$ -CDS problem for arbitrary constants  $k$  and  $m$  with  $m \geq k$ , which achieves performance ratio  $(2k - 1)\alpha_0$ , where  $\alpha_0$  is the performance ratio for the minimum CDS problem. Using the best known ratio  $\alpha_0 = 2 + \ln(\Delta + m - 2)$  for a general graph, our algorithm has performance ratio  $O(\ln \Delta)$ . To the best of our knowledge, this is the first approximation algorithm with a guaranteed performance ratio for  $k \geq 4$  in a general graph. For unit disk graph, using the best known ratio  $\alpha_0 = 1 + \varepsilon$ , where  $\varepsilon$  is an arbitrary real number in  $(0, 1)$ , our algorithm achieves a constant performance ratio of  $(2k - 1 + \varepsilon)$ , which is smaller than the constant ratio in [9], [20] (it should be noticed that our algorithm only works for the unweighted case while those in [9], [20] are valid for the weighted case).

In previous works [19], [31] on approximation algorithms for  $(k, m)$ -CDS in a general graph with  $k \leq 3$ , the main idea is to augment an  $(i, m)$ -CDS into an  $(i + 1, m)$ -CDS by extending a maximal  $(i + 1)$ -connected subgraph. However, for  $i \geq 3$ , such an idea has not succeeded in finding a performance guaranteed approximation because for  $i \geq 3$ , the changes of structures are very complicated when more nodes are added. In this paper, we use a new idea of extending a so-called  $i$ -block (the subgraph induced by which might even be disconnected).

The outline of our algorithm is as follows. The algorithm starts from a  $(1, m)$ -CDS  $C_0$  of  $G$  and increases its connectivity iteratively. We shall show that for any  $1 \leq i \leq k - 1$ , it is possible to make an  $(i, m)$ -CDS to be an  $(i + 1, m)$ -CDS in polynomial time by adding at most  $2|C_0|$  nodes. Unlike previous works which makes use of *brick decomposition* [32], [19], [31], the strategy of this paper is to expand a so-called  $i$ -block step by step, where an  $i$ -block

graph	$(k, m)$	ratio	reference
general	$(1, m)$	$2H(\Delta + m - 1)$	[30]
general	$(1, m)$	$2 + \ln(\Delta + m - 2)$	[32]
general	$(2, m)$	$2 + \alpha_0 + 2 \ln \alpha_0$ where $\alpha_0$ is performance ratio for $(1, m)$ -CDS	[19]
general	$(3, m)$	$\begin{cases} \alpha_1 + 8 + 2 \ln(2\alpha_1 - 6), & \text{for } \alpha_1 \geq 4 \\ 3\alpha_1 + 2 \ln 2, & \text{for } \alpha_1 < 4 \end{cases}$ where $\alpha_1$ is performance ratio for $(2, m)$ -CDS	[31]
general	$(k, m)$	$(2k - 1)\alpha_0$ where $\alpha_0$ is performance ratio for $(1, m)$ -CDS	*
UDG	$(2, 1)$	72	[24]
UDG	$(1, m)$	$\begin{cases} 5 + 5/m, & m \leq 5 \\ 7, & m > 5 \end{cases}$	[18]
UDG	$(2, m)$	$\begin{cases} 15 + 15/m, & 2 \leq m \leq 5 \\ 21, & m > 5 \end{cases}$	[18]
UDG	$(2, m)$	$\begin{cases} 7 + 5/m + 2 \ln(5 + 5/m), & \text{for } 2 \leq m \leq 5 \\ 12.89, & \text{for } m > 5 \end{cases}$	[19]
UDG	$(3, m)$	constant (280 for $m = 3$ )	[25]
UDG	$(3, m)$	$5\alpha_1^U$ , where $\alpha_1^U$ is performance ratio for $(2, m)$ -CDS on UDG (62.5 for $m = 3$ )	[26]
UDG	$(3, m)$	$\begin{cases} 26.34, & m = 3 \\ 25.68, & m = 4 \\ 26.86, & m \geq 5 \end{cases}$	[31]
UDG	$(k, m)$	minum weight version: constant	[9], [20]

TABLE I

RESULTS ON  $(k, m)$ -CDS WITH GUARANTEED PERFORMANCE RATIO. IN THIS TABLE,  $m \geq k$ . \* INDICATES THE RESULT OBTAINED IN THIS PAPER.

is a maximal set of nodes which cannot be separated by any  $i$ -separator of the subgraph induced by current backbone nodes [14]. Notice that the subgraph induced by an  $i$ -block can

even be disconnected. However, we can show that by adding at most  $2|C_0|$  nodes, an  $i$ -block can be expanded into an  $(i + 1)$ -connected graph. This is achieved by showing that as long as current backbone nodes do not induce an  $(i + 1)$ -connected graph, it is always possible to find at most two nodes in polynomial time, the addition of which strictly expands an  $i$ -block  $B$  by merging at least one node from  $C_0 \setminus B$ . Such expansion can be executed at most  $|C_0|$  times, resulting in the addition of at most  $2|C_0|$  nodes. Then, the desired performance ratio follows.

The paper is organized as follows. In Section II, some preliminary results are given. Section III presents the algorithm and analyzes its performance ratio. Section IV concludes the paper and points out some directions for future work.

## II. PRELIMINARY RESULTS

First, we introduce some notations used in this paper. For a node  $u \in V(G)$  and a node set  $C \subseteq V(G)$ ,  $N_C(v)$  is the set of nodes in  $C$  which are adjacent with  $u$  in  $G$ . In particular,  $N_G(u) = N_{V(G)}(u)$  is the *neighbor set* of  $u$  in  $G$ . For a node set  $U \subseteq V(G)$ ,  $N_G(U) = (\bigcup_{u \in U} N_G(u)) \setminus U$  is the *open neighbor set* of  $U$ , and  $N_G[U] = N_G(U) \cup U$  is the *closed neighbor set* of  $U$ . Suppose  $G$  is a connected graph. A node set  $S \subseteq V(G)$  is called a *separator* of  $G$  if  $G - S$  is no longer connected. In particular, a separator of  $G$  with cardinality  $i$  is called an  $i$ -separator of  $G$ .

The following result is a classic result in graph theory [1].

**Lemma II.1.** *Suppose  $G_1$  is an  $i$ -connected graph and  $G_2$  is obtained from  $G_1$  by adding a new vertex  $u$  and joining  $u$  to at least  $i$  vertices of  $G_1$ . Then  $G_2$  is also  $i$ -connected.*

**Corollary II.2.** *Suppose  $i$  and  $m$  are two positive integers with  $m \geq i + 1$ ,  $G$  is an  $(i + 1)$ -connected graph, and  $C$  is an  $(i', m)$ -CDS of  $G$  with  $i' \leq i$ . For any node set  $U \subseteq V(G) \setminus C$ ,*

- (i) node set  $C \cup U$  is also an  $(i', m)$ -CDS of  $G$ ;*
- (ii) if  $S$  is a separator of  $G[C \cup U]$  with cardinality at most  $i$ , then  $S \cap C$  is also a separator of  $G[C]$ ;*
- (iii) if  $G[C]$  is  $i$ -connected and  $S$  is an  $i$ -separator of  $G[C \cup U]$ , then  $S \subseteq C$  and  $S$  is also an  $i$ -separator of  $G[C]$ .*

*Proof:* Property (i) is the result of Lemma II.1 and the monotonicity of  $m$ -fold domination.

(ii) Suppose  $S \cap C$  is not a separator of  $G[C]$ . Then  $G[C] - (S \cap C)$  is connected. Since  $C$  is an  $m$ -fold dominating set, any node  $u \in U \setminus S$  has  $|N_C(u)| \geq m \geq i + 1 > |S|$ , and thus  $u$  has a neighbor in  $C \setminus S$ . It follows that  $G[C \cup U] - S$  is connected, contradicting that  $S$  is a separator of  $G[C \cup U]$ .

(iii) is a direct consequence of (ii). ■

Property (iii) implies that adding nodes into an  $(i, m)$ -CDS will not create new  $i$ -separators.

For a node set  $A$ , if  $V \setminus N_G[A] \neq \emptyset$ , then  $N_G(A)$  is a separator. In particular, if  $N_G(A)$  is a minimum separator of  $G$ , then  $A$  is called a *fragment* of  $G$ . The following inequality is well known (one may find it, for example, in [1]): for any two node sets  $A$  and  $B$ ,

$$|N_G(A)| + |N_G(B)| \geq |N_G(A \cap B)| + |N_G(A \cup B)|. \quad (1)$$

**Lemma II.3.** *Suppose  $A_1, A_2$  are two fragments of  $G$ . If  $A_1 \cap A_2 \neq \emptyset$  and  $V \setminus (N_G[A_1] \cup N_G[A_2]) \neq \emptyset$ , then both  $A_1 \cap A_2$  and  $A_1 \cup A_2$  are also fragments of  $G$ .*

*Proof:* By the condition of this lemma, both  $N_G(A_1 \cap A_2)$  and  $N_G(A_1 \cup A_2)$  are separators of  $G$ . Suppose the connectivity of  $G$  is  $i$ , then  $|N_G(A_1 \cap A_2)| \geq i$  and  $|N_G(A_1 \cup A_2)| \geq i$ . By the definition of fragment,  $|N_G(A_1)| = |N_G(A_2)| = i$ . Then by inequality (1),

$$2i = |N_G(A_1)| + |N_G(A_2)| \geq |N_G(A_1 \cap A_2)| + |N_G(A_1 \cup A_2)| \geq 2i, \quad (2)$$

which implies that  $|N_G(A_1 \cap A_2)| = |N_G(A_1 \cup A_2)| = i$ . The lemma follows. ■

For two node sets  $S, B \subseteq V(G)$ , we say that  $S$  cannot not separate  $B$  if  $G[B \setminus S]$  is connected. In the following, we give the definition of  $i$ -block, which was first studied by Mader [14], and receives a lot of attention recently because they offer a meaningful notion of the “ $k$ -connected pieces” into which the graph may be decomposed [3]. This notion is related to, but not the same as, the notion of a tangle proposed by Robertson and Seymour [16] in the study of minor.

**Definition II.4** ( $i$ -block). *Suppose  $G$  is an  $i$ -connected graph. A node set  $B$  is an  $i$ -block of  $G$  if  $B$  is a maximal node set satisfying the following two properties:*

- (i)  $|B| \geq i + 1$ , and
- (ii) *there is no  $i$ -separator of  $G$  which can separate  $B$ .*

By Menger’s Theorem, property (ii) in the above definition is equivalent to say that every pair of nodes in  $B$  are connected by at least  $(i + 1)$  internally disjoint paths in  $G$ . Notice that

the subgraph of  $G$  induced by an  $i$ -block  $B$  might even be disconnected. Those internally disjoint paths are required to be in  $G$ , not in  $G[B]$ . An example of a 3-block is given in Fig.1.

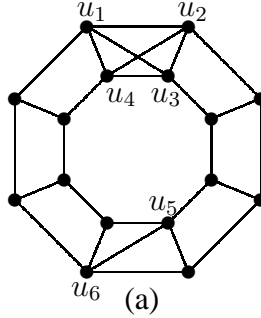


Fig. 1. Node set  $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$  forms a 3-block. Notice that  $G[U]$  is even disconnected. Also notice that  $U' = \{u_1, u_2, u_3, u_4\}$  is not a 3-block because it is not maximal with respect to the second property in Definition II.4.

We say that  $U$  is a *connected nodes set* of  $G$  if  $G[U]$  is connected.

**Lemma II.5.** Suppose  $i, m$  are two positive integers with  $m \geq i + 1$ ,  $C$  is an  $(i, m)$ -CDS of  $G$ ,  $U \subseteq V(G) \setminus C$  is a connected node set of  $G$ . Then no  $i$ -separator of  $G[C \cup U]$  can separate node set  $U \cup N_C(U)$ .

*Proof:* Consider an arbitrary  $i$ -separator  $S$  of  $G[C \cup U]$ . By Corollary II.2 (iii),  $S \subseteq C$ . In other words,  $S \cap U = \emptyset$ . Combining this with the assumption that  $G[U]$  is connected, we see that subgraph  $G[U \cup N_C(U)] - S$  is connected. ■

### III. ALGORITHM AND ITS ANALYSIS

#### A. The Highest Level Outline of Our Algorithm

Our algorithm starts from a  $(1, m)$ -CDS of  $G$ , and then augments its connectivity iteratively by adding more nodes. Suppose the  $(1, m)$ -CDS is  $C_0$ , which can be found using an existing algorithm such as the one in [32]. For  $i = 1, 2, \dots, k-1$ , in the  $i$ -th iteration, suppose an  $(i, m)$ -CDS is already at hand, say  $C$ , we shall show that it is possible to find at most  $2|C_0|$  nodes in polynomial time, such that adding them into  $C$  will result in an  $(i+1, m)$ -CDS. As a consequence, a  $(k, m)$ -CDS can be obtained in polynomial time whose size is at most  $(2k-1)|C_0|$ . Then, the desired performance ratio follows from the upper bound of  $|C_0|$  in [32].

### *B. Augmenting an $(i, m)$ -CDS into an $(i + 1, m)$ -CDS*

In the following, we focus on how to augment an  $(i, m)$ -CDS into an  $(i + 1, m)$ -CDS, where  $i < k$ . The idea is to expand an  $i$ -block iteratively until an  $(i + 1, m)$ -CDS is obtained. To be more concrete, suppose  $C$  is an  $(i, m)$ -CDS, and  $B$  is an  $i$ -block of  $G[C]$ . We shall show that as long as  $G[C]$  is not  $(i + 1)$ -connected, it is possible to find a node set  $U$  in polynomial time such that  $|U| \leq 2$ ,  $B$  is contained in a *strictly bigger*  $i$ -block of  $G[C \cup U]$ , say  $B'$ , and  $B' \setminus B$  contains at least one node of  $C_0$ . If  $G[C \cup U]$  is still not  $(i + 1)$ -connected, then we set  $C \leftarrow C \cup U$ ,  $B \leftarrow B'$ , and repeat. Because in each iteration, at least one node of  $C_0$  is newly merged into the expanded  $i$ -block, such a process can be executed at most  $|C_0|$  times. When it terminates, we must have an  $(i + 1)$ -connected subgraph, and the number of nodes added is at most  $2|C_0|$ .

To realize the above idea, one question is: what if  $G[C]$  does not contain an  $i$ -block. We shall deal with this problem later, and first deal with the case when  $G[C]$  already contains an  $i$ -block.

1) *Expanding an  $i$ -Block:* Suppose  $C$  is an  $(i, m)$ -CDS and  $G[C]$  is not  $(i + 1)$ -connected. Let  $S_0$  be an  $i$ -separator of  $G[C]$ . Since  $B$  is an  $i$ -block which cannot be separated by any  $i$ -separator,  $B \setminus S_0$  is contained in one connected component of  $G[C] - S_0$ , denote this connected component as  $G_1^{S_0}$ , and let  $G_2^{S_0}$  be the union of the other connected components of  $G[C] - S_0$  (see Fig.2(a)). By Corollary II.2 (ii),  $S_0 \cap C_0$  is a separator of  $G[C_0]$ . Let  $G_1$  be the union of those connected components of  $G[C_0] - (S_0 \cap C_0)$  which have nonempty intersections with  $G_1^{S_0}$ , and let  $G_2$  be the union of those remaining connected components of  $G[C_0] - (S_0 \cap C_0)$ . We claim that neither  $V(G_1)$  nor  $V(G_2)$  is empty. In fact, if  $V(G_2)$  is empty, then  $C_0$  is contained in  $V(G_1^{S_0}) \cup S_0$ . Consider a node  $u \in V(G_2^{S_0})$ , it is adjacent with at least  $m \geq k > i = |S_0|$  nodes of  $C_0$ , and thus has at least one neighbor in  $V(G_1^{S_0})$ , contradicting that  $G_1^{S_0}$  and  $G_2^{S_0}$  are disconnected by the removal of  $S_0$ . The case when  $V(G_1)$  is empty can be argued similarly.

Since  $G$  is  $k$ -connected and  $|S_0| = i < k$ , graph  $G - S_0$  is connected. Let  $P$  be a shortest path  $P$  between  $G_1$  and  $G_2$  in  $G - S_0$ . Suppose  $P = u_0 u_1 \dots u_t$ , where  $u_0 \in V(G_1)$  and  $u_t \in V(G_2)$ . We claim that  $t \leq 3$ . In fact, if  $t \geq 4$ , consider node  $u_2$ . Since  $C_0$  is an  $m$ -fold dominating set with  $m \geq k > |S_0|$ , node  $u_2$  is adjacent with some node  $u \in C_0 \setminus S_0$ . If  $u \in V(G_1)$ , then  $u u_2 \dots u_t$  is a shorter path between  $G_1$  and  $G_2$ . If  $u \in V(G_2)$ , then  $u_0 u_1 u_2 u$



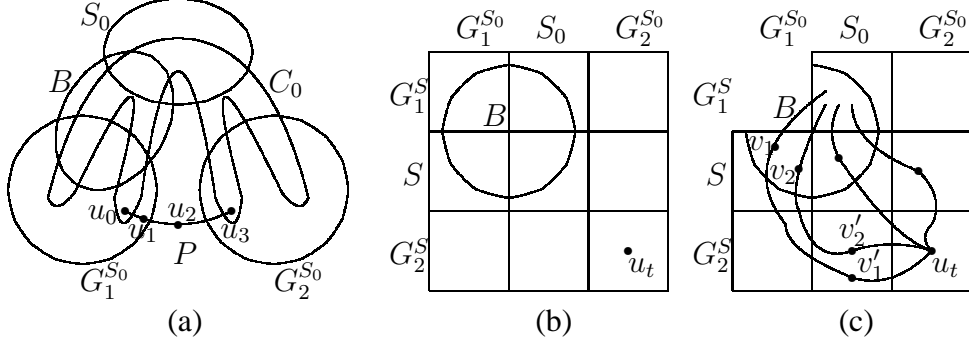


Fig. 2. (a) An illustration of how to find node set  $U$  to be added. The starfish-shaped region represents  $C_0$ . (b) The distribution of  $B$  and  $u_t$ . (c) An illustration for the proof of property  $(P_3)$ .

is a shorter path between  $G_1$  and  $G_2$ . Both contradict the choice of  $P$ . As a consequence, if we set  $U$  to be the set of internal nodes of  $P$ , then  $|U| \leq 2$ .

Notice that  $u_t \in C_0 \setminus B$ . So, if

$$\text{no } i\text{-separator of } G[C \cup U] \text{ can separate } B \cup \{u_t\}, \quad (3)$$

then we have found a desired  $U$  and may continue with the next iteration. Otherwise, there exists an  $i$ -separator  $S$  of  $G[C \cup U]$  which separates  $u_t$  from a node of  $B$ . Since  $B \setminus S$  is connected in  $G[C] - S$ ,

$$S \text{ separates } u_t \text{ from all nodes of } B \setminus S. \quad (4)$$

In the following, we shall consider nodes in  $B \setminus S$  as a whole.

By Corollary II.2 (iii),  $S$  is an  $i$ -separator of  $G[C]$ . Let  $G_1^S$  be a connected component of  $G[C] - S$  which contains  $B \setminus S$  (recall that  $B$  cannot be separated by  $S$ ). We have the following properties.

$(P_1)$  The distribution of  $B$  is as in Fig.2(b), since  $B \cap V(G_2^{S_0}) = \emptyset$  and  $B \cap V(G_2^S) = \emptyset$ .

$(P_2)$  Node  $u_t \in V(G_2^{S_0}) \cap V(G_2^S)$ . The reason why  $u_t \in V(G_2^{S_0})$  is by the choice of path  $P$ . The reason why  $u_t \in V(G_2^S)$  is because of (4).

$(P_3)$   $V(G_1^{S_0}) \cap V(G_1^S) \neq \emptyset$ . Suppose this is not true, then the distribution of  $B$  is as in Fig.2(c). It follows that  $B \cap V(G_1^{S_0}) \subseteq S$ . Since  $|B| \geq i + 1 > |S_0|$ , we have

$$|B \cap V(G_1^{S_0})| = |B \setminus S_0| > |S_0 \setminus B|. \quad (5)$$

Since  $S$  is an  $i$ -separator of  $G[C]$  separating  $u_t$  from  $B \setminus S$ , there are  $i$  internally disjoint paths in  $G[C]$  connecting  $B \setminus S$  and  $u_t$ , each path containing exactly one node of  $S$ . For a node  $v \in S$ , denote by  $P_v$  such a path containing  $v$ . For each node  $v \in B \cap V(G_1^{S_0}) \subseteq S$ , the path  $P_v$  must go through a distinct node  $v' \in S_0 \setminus B$  (see Fig.2(c)). But this is impossible because of (5). Property  $(P_3)$  is proved.

Taking  $A_1 = V(G_1^{S_0})$ ,  $A_2 = V(G_1^S)$ . Property  $(P_3)$  shows that  $A_1 \cap A_2 \neq \emptyset$ . Property  $(P_2)$  shows that  $u_t \in V(G_2^{S_0}) \cap V(G_2^S) = C \setminus (N_G[A_1] \cup N_G[A_2]) \neq \emptyset$ . So by Lemma II.3, the neighbor set of  $V(G_1^{S_0}) \cap V(G_1^S)$ , denoted as  $S_1$ , is also an  $i$ -separator of  $G[C]$ . Let  $G_1^{S_1}$  be the connected component of  $G[C] - S_1$  which contains  $B \setminus S_1$  (the left-top square of Fig.2(b) is  $G_1^{S_1}$ ). Notice that  $S \cap V(G_1^{S_0}) \neq \emptyset$ , otherwise  $B$  cannot be separated from  $u_0$  by  $S$ , and thus cannot be separated from  $u_t$  by  $S$  (since  $S \cap U = \emptyset$ ,  $u_0$  is connected to  $u_t$  through  $U$  in  $G[C \cup U] - S$ ). It follows that  $|V(G_1^{S_1})| < |V(G_1^{S_0})|$ .

To sum up, if for  $S_0$ , the shortest path  $P$  found in the above way does not satisfy property (3), then another  $k$ -separator  $S_1$  of  $G[C]$  can be found such that  $G_1^{S_1}$  is strictly smaller than  $G_1^{S_0}$ . Replace  $S_0$  by  $S_1$  and repeat, such a procedure can be executed at most  $|V(G_1^{S_0})| < n$  times. When it terminates, we are in the situation that property (3) is satisfied, at which time a desired node set  $U$  is found, and we may continue with the next iteration.

2) *When There Is No  $i$ -Block:* Now, we consider the situation that there is no  $i$ -block in  $G[C]$ . Suppose  $C$  is an  $(i, m)$ -CDS and  $G[C]$  is not  $(i + 1)$ -connected. Let  $S_0$  be an  $i$ -separator of  $C$ . By Corollary II.2 (ii),  $S_0 \cap C_0$  is a separator of  $G[C_0]$ . Let  $G_1$  be a connected component of  $G[C_0] - (S_0 \cap C_0)$  and let  $G_2$  be the union of the other connected components of  $G[C_0] - (S_0 \cap C_0)$ . Similarly to the above, a shortest path  $P$  between  $G_1$  and  $G_2$  in  $G - S_0$  has at most two internal nodes. Let  $U$  be the set of internal nodes of  $P$ . By Lemma II.5,  $U \cup N_C(U)$  cannot be separated by any  $i$ -separator of  $G[C \cup U]$ . Furthermore, since  $C$  is an  $m$ -fold dominating set of  $G$ , we have  $|N_C(U)| \geq m \geq k$ . So,  $|U \cup N_C(U)| \geq k + 1 > i + 1$ . In other words,  $U \cup N_C(U)$  is contained in an  $i$ -block of  $G[C \cup U]$ . This  $i$ -block can serve as the starting point of the subsequent iterations.

3) *To Sum Up:* The algorithm is summarized in Algorithm 1. The variable *flag* is used to indicate whether we should adjust the  $i$ -separator  $S_0$  used in the algorithm such that adding the internal nodes of the path can strictly expand current  $i$ -block  $B$  by at least one node of  $C_0$ . As we have shown in the above, the inner while loop is executed  $O(n)$  times, and the outer while loop is executed at most  $|C_0| = O(n)$  times. Since determining whether two

nodes can be separated by an  $i$ -separator can be done in polynomial time using a maximum flow algorithm, an  $i$ -block can be found in polynomial time (if it exists). In fact, this can be accomplished more efficiently using the method in [2], in which a stronger result was obtained by Carmesin *et al.* showing that there is an  $O(\min\{i, \sqrt{n}\} \cdot m \cdot n^2)$ -time algorithm which can find all  $i$ -blocks of a graph. All other operations can clearly be done in polynomial time. So, our algorithm has polynomial running time. The performance ratio of the algorithm is already implied in the above analysis.

**Theorem III.1.** *Suppose  $k, m$  are two positive integers with  $m \geq k$  and  $G$  is a  $k$ -connected graph. A  $(k, m)$ -CDS of  $G$  can be found in polynomial time whose size is at most  $(2k - 1)\alpha_0 \cdot \text{opt}$ , where  $\text{opt}$  is the size of an optimal solution and  $\alpha_0$  is the performance ratio for  $(1, m)$ -CDS of  $G$ . Taking  $\alpha_0 = 2 + \ln(\Delta + m - 2)$  which was obtained in [32], the performance ratio of our algorithm is  $(2k - 1)(2 + \ln(\Delta + m - 2))$ , which is  $O(\ln \Delta)$  for fixed  $k$  and  $m$ . Based on a PTAS to find a  $(1, m)$ -CDS in a unit disk graph, our algorithm has performance ratio  $2k - 1 + \varepsilon$  for  $(k, m)$ -CDS in unit disk graph.*

#### IV. CONCLUSION

In this paper, we proposed the first approximation algorithm with a guaranteed performance ratio for the minimum  $(k, m)$ -CDS problem in a general graph for general  $k$  under the assumption that  $m \geq k$ . The performance ratio is  $(2k - 1)(\ln(\Delta + m - 2) + 2)$ . Prior to this work, we have obtained  $(\ln \Delta + o(\ln \Delta))$ -approximation algorithms for  $k \leq 3$  [19], [31], [32], the strategy of which is to expand an  $i$ -brick greedily, where an  $i$ -brick is a maximal  $(i + 1)$ -connected *induced* subgraph. The difficulty of generalizing such a strategy to deal with higher value of  $k$  is that for  $i \geq 3$ , even when one could obtain a similar decomposition result using  $i$ -bricks, it is not clear how the decomposition structure changes when more nodes are added. In this paper, we propose using another strategy by expanding an  $i$ -block instead of expanding an  $i$ -brick. This new strategy works for any constant  $k$ . However, its performance ratio is not as delicate as those for  $k \leq 3$ . An improvement on the coefficient before  $\ln \Delta$  remains to be further explored.

#### ACKNOWLEDGMENT

This research is supported by NSFC (61222201, 11531011), and Xingjiang Talent Youth Project (2013711011).

---

**Algorithm 1**

---

Input: Two positive integers  $k, m$  with  $m \geq k$  and a  $k$ -connected graph  $G$ .

Output: A  $(k, m)$ -CDS  $C$ .

```
1: Find a  $(1, m)$ -CDS  $C_0$ .
2:  $C \leftarrow C_0$ .
3: for  $i = 1$  to  $k - 1$  do
4:   if  $G[C]$  is not  $(i + 1)$ -connected and there is no  $i$ -block in  $G[C]$  then
5:     Use the method in Section III-B2 to find a node set  $U$ .
6:      $C \leftarrow C \cup U$ .
7:      $B \leftarrow$  the  $i$ -block of  $G[C \cup U]$  containing  $U \cup N_C(U)$ .
8:   end if
9:   while  $G[C]$  is not  $(i + 1)$ -connected do
10:     $flag \leftarrow 1$ .
11:    Let  $S_0$  be an  $i$ -separator of  $G[C]$ .
12:    while  $flag = 1$  do
13:      Use the method in Section III-B1 to find a path  $P = u_0 \dots u_t$  in  $G - S_0$ .
14:      Let  $U'$  be the set of internal nodes of  $P$ .
15:      if  $B \cup \{u_t\}$  cannot be separated by any  $i$ -separator of  $G[C \cup U']$  then
16:         $flag \leftarrow 0$ .
17:         $U \leftarrow U', C \leftarrow C \cup U$ .
18:         $B \leftarrow$  the  $i$ -block of  $G[C]$  containing  $B \cup \{u_t\}$ .
19:      else
20:        Let  $S$  be an  $i$ -separator of  $G[C \cup U']$  separating  $B \cup \{u_t\}$ .
21:         $S_0 \leftarrow$  the neighbor set of  $V(G_1^{S_0}) \cap V(G_1^S)$  in  $G[C]$ , where  $G_1^{S_0}$  and  $G_1^S$ 
        are the connected components defined in Section III-B1.
22:      end if
23:    end while
24:  end while
25: end for
```

---

## REFERENCES

- [1] J. A. Bondy, U. S. R. Murty, Graph Theory, Springer, New York, 2008.

- [2] J. Carmesin, R. Diestel, M. Hamann, F. Hundertmark,  $k$ -Blocks: a connectivity invariant for graphs, *SIAM J. Discrete Math.*, 28(4) (2014) 1876–1891.
- [3] J. Carmesin, R. Diestel, F. Hundertmark, and M. Stein, Connectivity and tree structure in finite graphs, *Combinatorica*, 34 (2014), 1C-35.
- [4] X. Cheng, X. Huang, D. Li, W. Wu, D.-Z. Du, A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks, *Networks*, 42(4) (2003), 202–208.
- [5] F. Dai, J. Wu, On constructing  $k$ -connected  $k$ -dominating set in wireless ad hoc and sensor networks, *Journal of Parallel and Distributed Computing*, 66(7) (2006), 947–958.
- [6] B. Das, V. Bharghavan, Routing in ad-hoc networks using minimum connected dominating sets, *IEEE International Conference on Communications*, Montreal, 1997, 376–380.
- [7] D.-Z. Du, P.-J. Wan, *Connected Dominating Set: Theory and Applications*, Springer, New York, 2012.
- [8] A. Ephremides, J. E. Wieselthier, D. J. Baker, A design concept for reliable mobile radio networks with frequency hopping signaling, *Proceedings of the IEEE*, 75(1) (1987) 56–73.
- [9] T. Fukunage, Constant-approximation algorithms for highly-connected multi-dominating sets in unit disk graphs (2015) arXiv:1511.09156.
- [10] S. Guha, S. Khuller, Approximation algorithms for connected dominating sets, *Algorithmica*, 20(4) (1998) 374–387.
- [11] M. Li, P.-J. Wan, F. Yao, Tighter approximation bounds for minimum CDS in wireless ad hoc networks, *ISAAC’09, LNCS*, 5878 (2009), 699–709.
- [12] Y. Li, M. Thai, F. Wang, C. Yi, P.-J. Wan, D.-Z. Du, On greedy construction of connected dominating sets in wireless networks, *Wireless Communications and Mobile Computing*, 5 (2005), 927–932.
- [13] Y. Li, Y. Wu, C. Ai, F. Beyah, On the construction of  $k$ -connected  $m$ -dominating sets in wireless networks, *J. Combinatorial Optimization* 23 (2012), 118–139.
- [14] W. Mader, Über  $n$ -fach zusammenhängende Eckenmengen in Graphen, *J. Combin. Theory Ser. B*, 25 (1978), 74–93.
- [15] Z. Nutov, Approximating minimum-cost connectivity problems via uncrossable bifamilies. *AcM Transactions on Algorithms*, 9(1) (2012), 417–426.
- [16] N. Robertson, P.D. Seymour, Graph minors. X. Obstructions to tree-decomposition, *J. Combin. Theory Ser. B*, 52 (1991), 153C190.
- [17] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, K. Ko, A greedy approximation for minimum connected dominating sets, *Theoretical Computer Science*, 329(1) (2004), pp. 325–330.
- [18] W. Shang, F. Yao, P.-J. Wan, X. Hu, On minimum  $m$ -connected  $k$ -dominating set problem in unit disc graphs, *Journal of combinatorial optimization*, 16(2) (2008), 99–106.
- [19] Y. Shi, Y. Zhang, Z. Zhang, W. Wu, A greedy algorithm for the minimum 2-connected  $m$ -fold dominating set problem, *Journal of Combinatorial Optimization* 31 (2016) 136–151.
- [20] Y. Shi, Z. Zhang, D.-Z. Du, Approximation algorithm for minimum weight fault-tolerant virtual backbone in unit disk graph, <http://arxiv.org/abs/1508.05515>, a preliminary version appears in *INFOCOM2015*, 1080–1085.
- [21] M. Thai, N. Zhang, R. Tiwari, X. Xu, On approximation algorithms of  $k$ -connected  $m$ -dominating sets in disk graphs, *Theoretical Computer Science* 385 (2007) 49–59.
- [22] P.-J. Wan, K. M. Alzoubi, O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, *INFOCOM’02*, 2002, 1597–1604.
- [23] P.-J. Wan, L. Wang, F. Yao, Two-phased approximation algorithms for minimum CDS in wireless ad hoc networks, *ICDCS’08*, Beijing, 2008, pp. 337–344.

- [24] F. Wang, M. Thai, D.-Z. Du, On the construction of 2-connected virtual backbone in wireless networks, *IEEE Trans. Wireless Communications*, 8 (2009), 1230–1237.
- [25] W. Wang, D. Kim, M. K. An, W. Gao, X. Li, Z. Zhang, W. Wu, On construction of quality fault-tolerant virtual backbone in wireless networks, *IEEE/ACM Transactions on Networking*, 21(5) (2013), 1499–1510.
- [26] W. Wang, B. Liu, D. Kim, D. Li, J. Wang, Y. Jiang, A better constant approximation for minimum 3-connected  $m$ -dominating set problem in unit disk graph using Tutte decomposition, *INFOCOM2015*, Hongkong, 2015, 1796–1804.
- [27] W. Wu, H. Du, X. Jia, Y. Li, S. C. H. Huang, Minimum connected dominating sets and maximal independent sets in unit disk graphs, *Theoretical Computer Science*, 352(1) (2006), 1–7.
- [28] Y. Wu, F. Wang, M. Thai, Y. Li, Constructing  $k$ -connected  $m$ -dominating sets in wireless sensor networks, In: *Military communications conference*, Orlando, FL, 2007.
- [29] Z. Zhang, X. Gao, W. Wu, D.-Z. Du, A PTAS for minimum connected dominating set in 3-dimensional wireless sensor networks, *Journal of Global Optimization*, 45(3) (2009), 451–458.
- [30] Z. Zhang, Q. Liu, D. Li, Two algorithms for connected  $r$ -hop  $k$ -dominating set, *Discrete Mathematics, Algorithms and Applications*, 1(4) (2009), 485–498.
- [31] Z. Zhang, J. Zhou, Y. Mo, D.-Z. Du, Performance-guaranteed approximation algorithm for fault-tolerant connected dominating set in wireless networks, presented in *INFOCOM2016*.
- [32] J. Zhou, Z. Zhang, W. Wu, K. Xing, A greedy algorithm for the fault-tolerant connected dominating set in a general graph. *Journal of Combinatorial Optimization*, 28(1) (2014), 310–319.